

ния. – 2015. – № 5; URL: www.science-education.ru/128-22251 (дата обращения: 21.10.2015).

РАЗРАБОТКА МЕТОДА ДИНАМИЧЕСКОГО ПОДКЛЮЧЕНИЯ РЕСУРСОВ

А.Е. Нагиев, И.А. Ботыгин
(г. Томск, Томский политехнический университет)
e-mail: andrew_nagiev09@mail.ru

DEVELOPMENT OF A METHOD OF DYNAMIC CONNECTION OF RESOURCES

A.E. Nagiyev, I.A. Botygin
(Tomsk, Tomsk Polytechnic University)

The method of computing load balancing in horizontally distributed computing systems is described. The method is based on multi-threading and parallel processing. Data processing in this method occurs with a different number of model connected calculators. The scheme of dynamic connection of resources to reduce peak loads is showed.

Keywords: multithreading, distributed system, load balancing, socket technology, parallel computing.

Введение. Достаточно очевидно, что организация облачных вычислений и особенно разработка высокопроизводительных вычислительных систем (суперкомпьютеров) базируется на распределенных вычислительных системах (РАС) [1]. Отличительной РАС является параллельная обработка данных с помощью большого числа вычислительных узлов. Преимуществом данного подхода является возможность быстрого наращивания производительности путем горизонтального масштабирования вычислительных узлов или узлов хранения.

Существует ряд проблем, с которыми связана разработка РАС, среди которых: отсутствие возможности задавать глобальное время для всей РАС; связь между узлами осуществляется не мгновенно, а со значительными задержками, что требует дополнительных мер по разработке программного обеспечения [2] и другие проблемы.

Балансировка нагрузки (БН) применяется для оптимизации выполнения распределённых вычислений [3]. БН позволяет максимально равномерно распределять вычислительную нагрузку на имеющихся в системе узлах. После передачи задачи на выполнения в РАС, программное обеспечение с помощью заданного алгоритма БН осуществляет декомпозицию задачи, т.е. разбиение ее на модули, которые, передаются в узлы РАС для последующей их обработки.

Метод динамического подключения ресурсов. В настоящей работе, в результате проведения программных экспериментов был разработан метод балансировки нагрузки в распределенных вычислительных системах – метод динамического подключения ресурсов (ДПР). Были проведены специальные программные эксперименты, чтобы оценить эффективность разработанного метода при горизонтальном масштабировании вычислителей и увеличении нагрузки.

При проведении программного эксперимента сначала проводилась оценка эффективности классического серверного решения без балансировки нагрузки. Затем происходило сравнение нагрузочной способности этой системы с нагрузочной способностью системы, взаимодействие элементов которой происходило по методу ДПР.

По результатам проведенного эксперимента на основе вычислительной системы, состоящей из сервера управления и динамически подключаемых к нему терминалов запросов от имитаторов, показал, что при повышении количества поступающих запросов на обработ-

ку происходил отказ всей системы. При проведении эксперимента с помощью 4-х ядерного компьютера и имитации запросов в отдельных потоках по закону равномерного распределения с интервалом в 10 мс отказ происходил уже при обработке 850 запросов. Исходя из этого, можно сделать вывод о том, что при повышении частоты генерации запросов снижается быстродействие системы.

В связи с этим, чтобы повысить быстродействие выполнения задач был использован метод ДПР. Такой подход позволил динамически подключать дополнительные вычислительные узлы, с помощью которых обеспечивалось параллельное вычисление различных задач сервера управления, что обеспечило его разгрузку.

Для практической демонстрации предложенного метода был проведен, описанный ниже, программный эксперимент. Для проведения эксперимента использовалась функциональная структура распределенной системы, представленная на рис.1.



Рис. 1. Структурная схема моделируемой распределенной вычислительной системы

Основные изменения произошли с сервером обработки, который был заменен основным центром управления (ОЦУ). Также были добавлены динамически подключаемые диспетчеры нагрузки, к которым подключаются в отдельных потоках дополнительные центры управления (ДЦУ). Имитатор запросов остался без изменений. В отличие от предыдущего программного эксперимента, адрес соединения для каждого терминала запросов записывается в буфер, который удаляет подключение после обработки запроса.

С увеличением количества поступающих запросов ОЦУ, не успевая своевременно производить их вычисление, вызывает накопление запросов в буфере. При достижении заданного программно определенного количества запросов, ожидающих обработку в ОЦУ, начинает работу диспетчер распределения нагрузки, который запускает ДЦУ. Их количество ограничивается скоростью поступающих запросов и наличием дополнительных вычислительных мощностей в системе.

Каждый из запускаемых диспетчеров работает в своем отдельном потоке. Количество диспетчеров ограничивается только вычислительной мощностью инфраструктуры распределенной системы. Диспетчер выбирает определенное количество запросов из буфера и направляет их в ДЦУ, которые он запустил также в отдельных потоках. ДЦУ отключается только тогда, когда обработал полученные от диспетчера запросы.

Заключение. Эксперимент, описанный выше, показал эффективность разработанного метода динамического подключения ресурсов. Благодаря моделированию данной системы

можно сказать, что рассмотренный метод динамического подключения ресурсов останется эффективным и при значительном увеличении подключаемых вычислительных узлов.

ЛИТЕРАТУРА

1. Балансировка нагрузки: основные алгоритмы и методы // Хабрахабр. URL: <http://habrahabr.ru/company/selectel/blog/250201/> (дата обращения 26. 01. 2016)
2. Э. Таненбаум, М. ван Стеен. Распределенные системы. Принципы и парадигмы — СПб.: Питер, 2003. — 877 с:
3. Исследование стратегий балансировки нагрузки в системах распределенной обработки данных // КиберЛеника. URL: <http://cyberleninka.ru/article/n/issledovanie-strategiy-balansirovki-nagruzki-v-sistemah-raspredelennoy-obrabotki-dannyh> (дата обращения 03.02.2016)

ОБЗОР АЛГОРИТМОВ ДЛЯ СЕРВИСА РЕКОМЕНДАЦИИ КНИГ

К.С. Никитина

(г. Томск, Национальный исследовательский Томский политехнический университет)

e-mail: ksn1@tpu.ru

REVIEW ALGORITHMS FOR SERVICE RECOMMENDATIONS BOOKS

K.S. Nikitina

(Tomsk, National Research Tomsk Polytechnic University)

Abstract. This article describes recommendation systems methods and algorithms. It also it describes advantages and disadvantages of each method, analysis of similar recommendation services like service under development. The article describes the choice of recommendations algorithm is for the developing service.

Keywords: system recommendations, service, collaborative filtering, user-based collaborative filtering, item-based collaborative filtering.

Введение. Каждый человек оказывается в ситуации, когда ему необходимо что-то выбрать. Например, человек хочет посмотреть фильм на выходных, но не знает какой, а в его распоряжении есть только название, краткий сюжет и сопутствующий трейлер – это не простая задача.

Анализ методов рекомендации. Чтобы помочь пользователю выбирать художественные книги разрабатывается сервис рекомендации. Основной задачей любой рекомендательной системы является получение списка наиболее интересных объектов для определенного пользователя. Все системы рекомендаций можно разделить на три класса [1]:

- методы коллаборативной фильтрации;
- методы, анализирующие содержимое объектов;
- методы, основанные на знаниях.

Коллаборативные методы основаны на изучении поведения пользователя, то есть отслеживают оцененные объекты или действия. Процесс сбора информации о пользователе бывает явный и неявный. При явном процессе сбора пользователю предлагается оценить определенные объекты, а при неявном – программа сама отслеживает действия пользователя, например, количество просмотров одного и того же видеоролика. Рассмотрим алгоритм, который состоит из трех шагов: для каждого пользователя вычисляем, насколько его интересы совпадают с интересами конкретного пользователя, затем выбираем ближайших и, наконец, предсказываем оценку на основе оценок соседей с предыдущего шага. Описанный алгоритм